****

数据库原理

实 验 报 告

|  |  |
| --- | --- |
| **学 院** | 网络空间安全学院 |
| **专 业** | 网络工程 |
| **班 级** | 18272412 |
| **学 号** | 18041618 |
| **学生姓名** | 廖越强 |
| **教师姓名** | 丁宏 |
| **完成日期** | 2020.12.10 |
| **成 绩** |  |
| **实验一 利用SSMS创建与管理数据库** | | |
| 1. **实验目的**   掌握SQL Sever的视图、触发器和存储过程的使用 | | |
| 1. **实验内容**   （1）视图的创建与应用  （2）触发器的创建与应用  （3）存储过程的创建与应用 | | |
| 1. **实验环境**   Windows10 64位系统  SQL Server 2008 Express | | |
| 1. **主要操作步骤及实验结果记录** 2. 利用SSMS创建视图     2、利用SSMS创建下列视图   1. 创建视图View\_score，包括学号、姓名、专业名称、学院名称、课程号、课程名、课程类型、开课学期，学分，教师号、教师名，职称，成绩等列内容。        1. 创建视图View\_score\_XY01和View\_score\_XY02 ，视图包含View\_score的所有列，但前者只有“计算机学院”学生，后者只有“网络空间安全学院”的学生。   计算机学院      网络空间安全学院       1. 创建视图View\_majoy，视图列包括专业号、专业名称、学院号、学院名称。        1. 创建视图View\_teacher，视图列包括教师号、教师名称，性别、职称、学院号、学院名称。        1. 创建视图View\_course，视图列包括课程号、课程名称，课程类型、开课学期、学分、专业号、专业名称、学院号、学院名称。        1. 创建视图View\_student，视图列包括学号、姓名，性别、生日、班级、年级、专业号、专业名称、学院号、学院名称。       4． 利用视图实现数据查询和统计   1. 查询全校的教师的工号、姓名、所在学院名      1. 查询授课教师的工号、姓名、讲授课程名      1. 查询 “数据结构”课程的学生的学号、姓名、课程名、成绩、任课教师名      1. 查询“计算机学院”所有学生的学号、姓名、课程名、成绩、任课教师名      1. 查询所有授课的“教授”的工号、姓名和授课课程名        1. 统计“数据库”课程的最高分、最低分和平均成绩        1. 查询“数据库”课程高于平均分的学生学号和成绩        1. 查询平均分最高的课程号、课程名、任课教师、平均成绩        1. 查询平均分高于75分的学号、姓名、平均成绩、专业名称、学院名称        1. 查询计算机学院平均分高于75分的学生学号、姓名、平均成绩、专业名称       4. 创建触发器   1. 操作场景：学生“王五”退学处理。 2. 数据准备    1. 在tab\_student表中添加一个学生记录，如:   S020，王五，男，1999-8-9, …”。     * 1. 在tab\_score表中添加该学生的选课记录和成绩。      * 1. 创建tab\_studen1和tab\_score1表结构，作为退学学生的信息备份。        1. 具体操作要求如下   设计一个删除tab\_student表中记录的触发器，如果删除的记录是“王五”，则将该学生记录插入tab\_student1表中，同时在tab\_score表中删除该学生选课记录，并将该学生的记录插入tab\_score1表中；否则禁止删除记录   1. 创建触发器   直接运行使用ppt里的命令行     1. 运行触发器 2. DELETE FROM tab\_student WHERE sname <> '王五'       tab\_student、tab\_student1、tab\_score和tab\_score1均没有受到影响   1. DELETE FROM tab\_student WHERE sname= '王五'     tab\_student、tab\_score和中有关王五的数据都被删除，tab\_student1、tab\_score1新增了王五的数据，如图  tab\_student、tab\_score      tab\_student1、tab\_score1      **5. 存储过程的创建与使用**  写出下列存储过程的创建语句和执行语句  1. 以下操作返回结果集   * 1. 根据姓氏查询学生信息（用函数实现）   USE stuinfo  IF EXISTS (SELECT name FROM sysobjects WHERE name = 'stu\_info\_name' AND type = 'P')  DROP PROCEDURE stu\_info\_name  GO  CREATE PROCEDURE stu\_info\_name  @sname varchar(10)  AS  SELECT \* FROM  tab\_student  WHERE sname LIKE @sname+'%'  GO  执行 EXECUTE stu\_info\_name '王'     * 1. 查询最近几天内过生日的学生信息   USE stuinfo  IF EXISTS (SELECT name FROM sysobjects WHERE name = 'Find\_Birthday\_StuInfo' AND type = 'P')  DROP PROCEDURE Find\_Birthday\_StuInfo  GO  CREATE PROCEDURE Find\_Birthday\_StuInfo  AS  SELECT \* FROM tab\_student  WHERE DATEDIFF(day, GETDATE(),  CAST(  CAST(YEAR(GETDATE()) as CHAR(4))  +'-'+  CAST(MONTH(birthday) as CHAR(2))  +'-'+  CAST(DAY(birthday) as CHAR(2))  as datetime  )  )>-6    GO  EXECUTE Find\_Birthday\_StuInfo     * 1. 根据课程名查询该课程的选课学生学号、姓名和成绩   USE stuinfo  IF EXISTS (SELECT name FROM sysobjects WHERE name = 'c\_info\_name' AND type = 'P')  DROP PROCEDURE c\_info\_name  GO  CREATE PROCEDURE c\_info\_name  @cname varchar(10)  AS  SELECT s.sno,sname,score FROM tab\_score sc INNER JOIN  tab\_student s ON s.sno = sc.sno INNER JOIN tab\_course c ON sc.cno = c.cno  WHERE cname = @cname  GO  EXECUTE c\_info\_name '数据库'     * 1. 查询某一学期所有课程的课程名、最高分、最低分和平均成绩   创建存储过程  USE stuinfo  /\* 创建视图 \*/  IF EXISTS (SELECT \* FROM sysobjects WHERE name = 'View\_score\_6\_4' AND type = 'V')  DROP VIEW View\_score\_6\_4  GO  CREATE VIEW View\_score\_6\_4  AS  SELECT a.sno, a.cno, cname,term,score  FROM tab\_score a  INNER JOIN tab\_teacher c ON a.tno = c.tno  INNER JOIN tab\_course d ON a.cno = d.cno  GO  /\* 创建存储过程 \*/  IF EXISTS (SELECT \* FROM sysobjects WHERE name = 'c\_score' AND type = 'P')  DROP PROCEDURE c\_score  GO  CREATE PROCEDURE c\_score  @term smallint  AS  SELECT cname AS 课程名,max(score)AS 最高分,min(score)AS 最低分,AVG(score) AS 平均成绩  FROM View\_score\_6\_4  WHERE term=@term GROUP BY cname  GO  EXECUTE c\_score 1    2. 以下操作以参数形式返回结果   * 1. 根据学院名称统计该学院的教师人数   USE stuinfo  /\*存储创建过程\*/  IF EXISTS (SELECT \* FROM sysobjects WHERE name = 'inst\_name' AND type = 'P')  DROP PROCEDURE inst\_name  GO  CREATE PROCEDURE inst\_name  @iname varchar(10),  @teacher\_num smallint OUTPUT  AS  SELECT @teacher\_num=COUNT(\*) FROM tab\_teacher t  inner join tab\_institute i on t.instno=i.instno  WHERE instname LIKE '%'+@iname+'%'  GO  /\* 存储过程执行 \*/  DECLARE @teacher\_num smallint  EXECUTE inst\_name '计算机学院', @teacher\_num OUTPUT  IF @teacher\_num is NULL  PRINT 'ERROR: No teacher!'  ELSE  PRINT 'The number of teachers is '+CAST(@teacher\_num AS char(3))   * 1. 根据课程名统计该课程的最高分   USE stuinfo  IF EXISTS (SELECT name FROM sysobjects WHERE name = 'c\_name' AND type = 'P')  DROP PROCEDURE c\_name  GO  CREATE PROCEDURE c\_name  @c\_name varchar(10),  @max\_score smallint OUTPUT  AS  SELECT @max\_score = MAX(score) FROM tab\_score sc INNER JOIN  tab\_course c ON sc.cno = c.cno  WHERE cname=@c\_name  GO  DECLARE @max\_score smallint  EXECUTE c\_name '数据库',@max\_score OUTPUT  IF @max\_score is NULL  PRINT 'ERROR: No score!'  ELSE  PRINT 'The max score is '+CAST(@max\_score AS char(3))     * 1. 根据学生姓名统计该学生已获得的学分   USE stuinfo  IF EXISTS (SELECT name FROM sysobjects WHERE name = 'Find\_Credit\_Of\_Stu' AND type = 'P')  DROP PROCEDURE Find\_Credit\_Of\_Stu  GO  CREATE PROCEDURE Find\_Credit\_Of\_Stu  @sname varchar(10),  @credits smallint OUTPUT  AS  SELECT @credits = SUM(credit) FROM View\_Score  WHERE sname=@sname  GO  DECLARE @credits smallint  EXECUTE Find\_Credit\_Of\_Stu '李四', @credits OUTPUT    IF @credits is NULL  PRINT 'No credits'  ElSE  PRINT 'This student has gained totally '+CAST(@credits as char(3))+'credits already.' | | |
| 1. **实验分析总结及心得**   （结合所学知识对实验过程中观察到的实验结果进行分析总结，以便加深对知识的理解，并总结通过实验学到的知识或技术）  1、本次实验我学到的知识：我学会使用SQL Server的GUI界面——SQL Server Management Studio(SSMS)来创建视图、修改视图，除此之外，我还学到了创建触发器、删除触发器、创建存储过程、存储过程的执行等知识。  2、本次实验我遇到的问题：在完成视图相关的任务时，发现简单的视图可以通过ssms快速建立，但是复杂一点的，比如需要求平均，然后输出高于或者低于平均值的这种，其实还是使用T-sql语句来写会更好理解一些。触发器的话，使用的时候总感觉不是很放心，因为它是自动触发的，如果出了什么意料之外的情况，感觉很难排查出来。而在完成存储这方面的任务时，对T-SQL有了更深的理解，T-SQL对我来说像是一门新的编程语言，它也有变量和函数，允许用户声明变量、有条件执行以及其它的编程功能，可包含程序流以及对数据库的查询。它可以接受参数，也可返回参数，还可返回单个或多个结果集。也了解到存储过程的一些优点：（1）单个存储过程中执行一系列 SQL 语句。（2）可以从自己的存储过程内引用其它存储过程，这可以简化一系列复杂语句。（3）存储过程在创建时即在服务器上进行编译，所以执行起来比单个 SQL 语句快。 | | |